OK to enter. 7/17/06

AMENDMENTS TO THE SPECIFICATION

Please amend paragraph [0008] beginning on page 4 as follows:

A match section 120 of the CAM cell 100 is comprised of transistors M1, M1#, M2, M2# and M3, which controllably couple the match line ML to ground in certain situations. In the CAM cell 100, a "no-match" condition is detected if the match line ML remains at a precharge potential ~~if the match line ML is pulled to ground during the match operation~~, which a "match" condition is detected ~~if the match line ML remains at a precharge potential~~ if the match line ML is pulled to ground during the match operation. Typically, the detection of the state of the match line ML is performed by a sense amplifier (not illustrated).

Please amend paragraph [0010] beginning on page 5 as follows:

Transistors M1, M1#, M2, and M2# form a comparison circuit.. The gates of transistors M2, M2# are respectively coupled to search data lines SD and SD#. The search data is placed on the line SD while the complement of the search data is placed on the line SD#. Similarly, the logical value stored at node D is coupled to the gate of transistor M1 while the complement of that logical value stored at node D# is coupled to the gate of transistor M1#. In this manner, the comparison structure will pull the match line ML voltage from its precharged level to ground if and only if the search data on line SD matches the data stored at node D [[and]] or the complement of the search data on line SD# matches the complement of the data stored at node D, assuming that transistor M3 is conducting. If transistor M3 is not conducting, the match line ML potential will remain at its pre-charged level ~~to force a "match" condition~~.

Please amend paragraph [0023] beginning on page 8 as follows:

FIG. 4 is a flowchart illustrating an exemplary stuck match line test S31 according to an embodiment of the invention. The objective of this test is to identify and mask for any stuck match lines to remove the stuck match lines from the CAM and replace them with a redundant match row. The test S31 begins at step S41, where the data storage location 110 of each CAM cell 100 is written with a 1-pattern. Execution continues at step S42, where the search mask M of each is set to a not mask state (to consider every cell to participate in match). (If device 210 is a binary CAM, there is no mask bit, so step S42 would not be performed. Rather, execution would continue at step S43 after S41.) This ensures that every CAM cell 100 will participate in the search by enabling the comparison portion 120 of each CAM cell 100 to pull down its associated match line in case of a ~~mismatch~~ match. At step S43, the CAM cell 100 is searched using a 0-pattern search data pattern. At step S44, an inquiry is made to see if there were any matches. In a correctly operating device 210, there should be no matches since step S41 set every data location to a logical one and the search data contained zeros. Thus, if there are no matches, the test completes. A match is an indication of a faulty match line. If there are multiple faulty match lines then the highest priority matched line will be the match address. The address line associated with the matching entry is therefore disabled (S45). If there is a redundant match line, it can be remapped to replace of the disabled match line. After step S45, execution resumes at step S43. Thus, steps S43, S44, and S45 are repeatedly performed in a loop until the test S31 ends with no matches or until all redundant resources are exhausted, when the CAM is declared as unrepairable. This process can be repeated with the CAM filled with 0-pattern and a 1-pattern as the match pattern. This pattern though not strictly necessary will be useful if the stuck match lines are sensitive to the patterns. The further process flows assumes that there is no 0-pattern was applied. At the end of the process therefore the memory will be left with a 1-pattern.

Please amend paragraph [0025] beginning on page 9 as follows:

The weak pull down test S32 begins at step S51, which, in cooperation with steps S55 and S56 sets up a for-loop iterating through steps S52-54, as described below. The index i is set to one during the first iteration and the loop terminates when i is incremented such that it exceeds the width of a word in the CAM device 210. At step S52, a search is performed using a search pattern. The search pattern is constructed so that each bit in the search pattern is the compliment of the corresponding 1-pattern, except for bit-i, which is different and is set to equal as corresponding pattern i. This pattern will be referred to as Walking-[[0]]1 pattern. At step S53, the search results are examined to determine whether there is a match. If a match was detected, the method S32 has detected an error and the error is managed in step S54.

Please amend paragraph [0029] beginning on page 10 as follows:

Steps S58, S62, and S63 form a for-loop much like steps S51, S55, and S56, as previously described. Execution begins with a first iteration at step S59, where a search is executed with a search expression set to have each bit compliment of 0-pattern except for bit-i, which is equal to the bit i. This pattern will be referred to as a Walking-[[I]]0 pattern. In step S60, if a match is found there is an error and the error is managed by executing step S61. Step S61 is identical to step S54 (previously described). If no match is detected at 560, the for-loop continues with steps S62 and S63, which operate similarly to steps S55 S56. The end of execution of the for-loop comprising steps S58, S62, and S63 ends the weak pull down test S32.